Journal of Engineering and Science Research 7 (2): 20-24, 2023
e-ISSN: 2289-7127
© RMP Publications, 2023
DOI: 10.26666/rmp.jesr.2023.2.4

A Review: Restricted Splicing Systems

Mathuri Selvarajoo [1,*], Mohd Pawiro Santono[1], Fong Wan Heng[2] , Nor Haniza Sarmin[2], Vincent Daniel David[1]

[1]School of Mathematical Sciences, College of Computing, Informatic and Media, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia
[2]Department of Mathematical Sciences, Universiti Teknologi Malaysia, 81310 UTM, Johor Bahru, Johor, Malaysia
*Corresponding Author: mathuri@tmsk.uitm.edu.my

Received: 14 February 2023; Revised: 15 March 2023; Accepted: 30 March 2023; Published: 30 April 2023

Abstract: In 1987, Head [1] proposed a splicing method as a mathematical model for DNA recombination. In this model, two DNA molecules are cut at specific recognition sites, and the prefix of one molecule is combined with the suffix of the other, creating a new string. Splicing operations in the system are represented as splicing rules, formalizing the process as a string operation. By iteratively applying a set of splicing rules to a set of initial strings or axioms, a language can be generated, which is known as a splicing language. According to the Chomsky hierarchy, these languages are classified as regular languages, the lowest level of language. To enhance the generative power of splicing languages, restrictions are introduced. This research reviews three splicing system restrictions: weighted splicing [2], group splicing [3], and probabilistic splicing [4].

Keywords: *Splicing Systems, Formal Language Theory, Splicing Languages*

Introduction

Every living organism possesses a unique DNA structure. The double-helical form of DNA was first introduced by Watson and Crick in 1953 [5]. DNA molecules are made up of nucleotides, which consist of three basic components: sugar, phosphate, and base [6]. The sequence of bases in DNA, namely Adenine, Guanine, Cytosine, and Thymine (abbreviated as A, G, C, and T), differs from one structure to another. These bases are linked together by hydrogen bonds in accordance with base-complementary rules, where A pairs with T and G pairs with C, forming the codes a, g, c, and t, respectively [5].

Head introduced splicing systems in 1987 [1] as a way to model the recombinant behavior of double-stranded DNA (dsDNA) and the enzymes responsible for cutting and pasting dsDNA. Restriction enzymes, which are naturally present in bacteria, can cleave DNA fragments at specific sequences known as restriction sites, while ligases can reconnect DNA fragments with complementary ends [6]. This model consists of a finite alphabet *V*, a finite set of initial strings over alphabet *A*, and a finite set of rules *R* that operate on the strings through iterative cutting and pasting, resulting in the generation of new strings [5].

A splicing language is a language that is generated by a splicing system. It has been proven that all splicing languages with finite sets of axioms and rules are regular. Therefore, to increase the generative power of splicing systems, researchers have investigated various restrictions on the use of rules, such as weighted [2], groups [3], probability [4], and more recently, fuzzy restrictions [7]. Additionally, restrictions have been introduced in variants of splicing systems, such as simple and semi-simple splicing systems [8-10]. All these restrictions serve a common purpose, which is to enhance the generative power of languages produced by splicing systems. This is particularly important in the field of DNA computing, where splicing systems with the highest generative power, which is recursively enumerable (RE), can be viewed as theoretical models for universally programmable DNA-based

Corresponding Author: Mathuri Selvarajoo, School of Mathematical Sciences, College of Computing, Informatic and Media, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia. Email: mathuri@tmsk.uitm.edu.my

computers.

This paper presents a review of the restrictions applied to splicing systems, including weighted, groups, and probability. The paper is structured as follows: Section 2 provides important definitions and notations from formal language theory and splicing systems. In Section 3, the concept of weighted splicing systems and their outcomes are discussed. Section 4 explores splicing systems over permutation groups and the languages produced by permutation groups of length two. Section 5 reviews probabilistic splicing systems and their results. Finally, in Section 6, the research is concluded with a discussion on the overall findings.

This section provides an overview of the fundamental concepts and notations from formal language and systems theories that will be utilized later in the paper. For further information on these topics, interested readers can refer to sources such as [11], [12], [13], [14].

The following general notations are used throughout the paper. The membership of an element to a set is denoted by $\in$, whereas the negative of set membership is denoted by $\notin$. The strictness of the inclusion is denoted by $\subseteq$ and $\subset$ stands for (proper) inclusion. The empty set is represented by the symbol $\varnothing$. | X | denotes the cardinality of a set X.

The family of recursively enumerable, context-sensitive, context-free, linear, regular, and finite languages are denoted by **RE, CS, CF, LIN, REG** and **FIN**, respectively. For these language families, the next strict inclusions, named *Chomsky hierarchy* (see [12]), hold:

Further, some basic definitions and results of iterative splicing systems were recalled. Let V be an alphabet, # and \$ two special symbols. A splicing rule over V is a string of the form

$$r = u_1 \# u_2 \# u_3 \# u_4 \text{ where } u_i \in V^*, 1 \le i \le 4.$$

For such a rule $r \in R$ and strings $x,y,z \in V^*$, $(x,y) \,\square_r\, z$ if and only if $x = x_1u_1u_2x_2$, $y = y_1u_3u_4y_2$, and $z = x_1u_1u_4y_2$, for some $x_1,x_2,y_1,y_2 \in V^*$.

The string z is said to be obtained by splicing x and y, as indicated by the rule r; the strings u_1u_2 and u_3u_4 are called the sites of the splicing. The first term called x and y is the second term of the splicing operation.

A H scheme (a splicing scheme) is a pair $\sigma = (V, R)$, where V is an alphabet and $R \subseteq V^*\#V^*\$V^*\#V^*$ is the set of the splicing rule. For a given H scheme $\sigma = (V,R)$ and a language $L \subseteq V^*$,

$$\sigma(L) = \{z \in V^* \mid (x,y) \,\square_r\, z \text{ for some } x,y \in L, r \in R,$$

are defined and iterative splicing languages are defined as

$$\sigma^*(L) = \bigcup_{i \ge 0} \sigma^i(L)$$

$$\sigma^0(L) = L\ ,$$

$$\sigma^{i+1}(L) = \sigma^i(L) \cup \sigma(\sigma^i(L)), i \ge 0.$$

An extended H system is a construct $\gamma = (V,T,A,R)$ where V is an alphabet, $T \subseteq V$ isa terminal alphabet, $A \subseteq V^*$ is the set of axioms, and $R \subseteq V^*\#V^*\$V^*\#V^*$ is the set of splicing rules. The system is said to be non-extended when $T = V$. An alphabet $x \in V$ is said to be non-terminal when $x \notin V$. The language generated by γ is defined by

$$L(\gamma) = \sigma^i(A) \cap T^*$$

The symbol EH (F_1, F_2) denotes the family languages generated by extending H system $\gamma = (V,T,A,R)$ with $A \in F_1$ and $R \in F_2$ where

$$F_1, F_2 \in \{\mathbf{FIN}, \mathbf{REG}, \mathbf{CF}, \mathbf{LIN}, \mathbf{CS}, \mathbf{RE}\}$$

The following theorem shows the relations of the family of languages generated by splicing systems to the families of Chomsky languages.

Theorem 2.1.

[15]: The relations in Table 2.1 hold, where at the intersection of the row marked with F_1 with the column marked with F_2 there appear either the family EH (F_1, F_2) or two families F_3, F_4 such that $F_3 \subset$ EH (F_1, F_2) $\subseteq F_4$:

$F_1 \setminus F_2$	FIN	REG	LIN	CF	CS	RE
FIN	**REG**	**RE**	**RE**	**RE**	**RE**	**RE**
REG	**REG**	**RE**	**RE**	**RE**	**RE**	**RE**
LIN	**LIN, CF**	**RE**	**RE**	**RE**	**RE**	**RE**
CF	**CF**	**RE**	**RE**	**RE**	**RE**	**RE**
CS	**RE**	**RE**	**RE**	**RE**	**RE**	**RE**
RE	**RE**	**RE**	**RE**	**RE**	**RE**	**RE**

TABLE 1. The family of languages generated by splicing systems.

3. Weighted Splicing System

This section covers the introduction of weighted splicing systems by Turaev et al. [2]. These systems are characterized by a weighting space and operations that are closed in that space. The concept of threshold languages generated by weighted splicing systems is also presented, and the results indicate that these systems can generate languages with greater generative power than regular languages. The definition of weighted splicing systems is provided as follows:

Further, weighted splicing operation and languages generated were defined:

In this paper, the sets of integers, positive rational numbers, the set of integers with Cartesian products, and the set of 2 × 2 matrices with integer entries are all taken into consideration as weighting spaces.

From the definition, the next lemma follows immediately.

The generative power of weighted splicing systems was demonstrated through an example in which various weighting spaces were used to generate strings with the same set of axioms and splicing rules. This example showed that the selection of weighting spaces has a significant impact on the generative power of the system.

Example 3.1. Consider a weighted splicing system $\gamma = (\{a, b, c, w, x, y\}, \{a, b, w\}, \{(wax, \tau_1), (xby, \tau_2), (ycw, \tau_3)\}, \{r_1 = a\#x\$w\#ax, r_2 = b\#y\$x\#by, r_3 = c\#w\$y\#cw, r_4 = a\#x\$x\#b, r_5 = b\#y\$y\#c\}, \omega, M, \odot)$.

For all $k,m,n \geq 1$,

$$(wa^k x, wax) \square r_1 wa^{k+1} x,$$
$$(xb^m y, xby) \square r_2 xb^{m+1} y,$$
$$(yc^n w, ycw) \square r_3 yc^{n+1} w$$

Further,

$$(wa^k x, xb^m y) \square r_4 wa^k b^m y, \text{for } k, m \geq 1,$$

and

$$(wa^k b^m y, yc^n w) \square r_5 wa^k b^m c^n w, \text{for } k, m, n \geq 1.$$

Then, the language generated by the weighted splicing system γ is

$$L_\omega(\gamma) = \{wa^k b^m c^n w \mid (wa^k b^m c^n w, \omega(wa^k b^m c^n w)) \in \sigma^*(A), k, m, n \geq 1\}$$

where $A = \{(wax, \tau_1), (xby, \tau_2), (ycw, \tau_3)\}$.

Next, different threshold languages with different weighting spaces and operations are defined.

First, let M $= \mathrm{Q}^+$, the operation $\odot$ be the usual multiplication, and $\tau_1 = 3^{-1}, \tau_2 = 5^{-1}, \tau_3 = 15$. Then,

$$L_\omega(\gamma) = \{wa^k b^m c^n w \mid (wa^k b^m c^n w, 3^{n-k} 5^{n-m}) \in \sigma * (A), k, m, n \geq 1\}$$

$\tau = 1$ was chosen as a cut-point, and define the following threshold languages:

$$L_\omega(\gamma, =1) = \{wa^k b^m c^n w \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF},$$
$$L_\omega(\gamma, >1) = \{wa^k b^m c^n w \mid n > k, m \geq 1\} \in \mathbf{CF} - \mathbf{REG},$$
$$L_\omega(\gamma, <1) = \{wa^k b^m c^n w \mid k, m > n \geq 1\} \in \mathbf{CF} - \mathbf{REG}.$$

Second, let M = $\mathrm{Z} \times \mathrm{Z}$, the operation $\odot$ is defined as the component wise addition of pairsfrom $\mathrm{Z} \times \mathrm{Z}$, and $\tau_1 = (1, 0)$, $\tau_2 = (-1, 1)$, $\tau_3 = (0, -1)$. Then,

$$L_\omega(\gamma) = \{wa^k b^m c^n w \mid (wa^k b^m c^n w, (k-m, m-n)) \in \sigma^*(A), k, m, n \geq 1\}.$$

Consequently,

$$L_\omega(\gamma, =(0,0)) = \{wa^k b^m c^n w \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF},$$
$$L_\omega(\gamma, >(0,0)) = \{wa^k b^m c^n w \mid k > m > n \geq 1\} \in \mathbf{CS} - \mathbf{CF},$$
$$L_\omega(\gamma, <(0,0)) = \{wa^k b^m c^n w \mid n > m > k \geq 1\} \in \mathbf{CS} - \mathbf{CF}.$$

From Lemma 3.1 and Example 3.1, the following results were obtained.

Theorem 3.1.

For $F_1 \in \{\mathbf{LIN}, \mathbf{CF}\}$,

$$\omega EN(\mathbf{FIN}, \mathbf{FIN}) - EN(F_1, \mathbf{FIN}) \neq \varnothing.$$

Theorem 3.2.

$$\mathbf{REG} \subset \omega EN(\mathbf{FIN}, \mathbf{FIN}) \subseteq \mathbf{RE}.$$

Research has demonstrated that the inclusion of weights in splicing systems, even with a simple extension, leads to an increase in their generative power beyond that of regular languages. However, the generative power of restricted splicing systems is yet to produce recursively enumerable languages, and therefore further research in this area remains an open field.

In 2012, Hamzah et al. presented splicing systems over permutation groups of length two [3]. These systems use the elements of permutation groups as valences to calculate the generative power of extended valence splicing systems over permutation groups. The following definition describes an extended valence H system over a permutation group.

For $(x,v_1),(y,v_2),(w,v_3) \in V^* S_n$ and $r \in R$, where $x,y,w \in V^*$, $v_1, v_2, v_3, \in S_n$, the splicingoperation is $[(x,v_1),(y,v_2)] \vdash_r (w,v_3)$ if and only if $(x,y) \vdash_r w$ and $v_3 = v_1 \cdot v_2$. Then $L(\gamma) = \{x \in T^* \mid (x,e) \in \sigma^*(A)\}$.

The computation of the group operation of new strings in extended valence H systems over permutation groups is done by associating an element of the permutation group to each axiom A and each splicing operation. If the computation of the associated elements of the group results in the identity element, then the complete strings produced are valid.

An example of splicing system over permutation group of length two involving one initial string is shown in the following.

Continuing this splicing process, the resulting language is only accepted if the value of valences is equal to the identity. Therefore, the language of this extended valence splicing system is $L(\gamma) = \{ca^{2n}d, n \geq 1\}$. From the Chomsky grammar, the grammars that generate this language are context-sensitive and context-free grammar but not regular.

The example presented demonstrates that splicing systems over permutation groups of length two with a single initial string can generate languages beyond regular languages, indicating an increase in generative power. However, such systems are still unable to generate recursively enumerable languages, indicating that while some permutation groups can increase the generative power of splicing systems up to context-sensitive, it remains an open question whether any splicing system can generate recursively enumerable languages.

5. Probabilistic Splicing Systems

In 2013, Mathuri et. al. [4] introduced probabilistic as a restriction of splicing systems.In this paper, probabilities are associated with the axioms, and the probability of the generated string from two strings is calculated by multiplication of their probabilities. The threshold probabilistic splicing languages were defined and showed that probabilistic systems with finite component can increase the generative power of the splicing languages generated. The definition of probabilistic splicing systems as follows:

From the definition before, the next lemmas follow immediately.

define the threshold language generated by γ' as $L_p(\gamma', > 0)$, then it is not difficult to see that $L(\gamma) = L_p(\gamma', > 0)$. □

An example illustrates that the use of thresholds with probabilistic systems increase generative power of splicing systems with finite components up to context-sensitive languages.

Example 5.1. Consider the probabilistic splicing system,

$$\gamma_2 = (\{a,b,c,w,x,y,z\},\{a,b,c,w,z\},A_2,R_2,p_2)$$

where $A_2 = \{(wax, 3/19), (xby, 5/19), (ycz, 11/19)\}$ and $R_2 = \{r_1 = wa\#x\$w\#a, r_2 = xb\#y\$x\#b, r_3 = yc\#z\$y\#c, r_4 = a\#x\$x\#b, r_5 = b\#y\$y\#c\}$.

Using the first axiom and rule r_1, obtain strings

$$(wa^k x, (3/19)^k), k \geq 1,$$

the second axiom and rule r_2,

$$(xb^m y, (5/19)^m), m \geq 1,$$

the third axiom and rule r_3,

$$(yc^n z, (11/19)^n), n \geq 1.$$

The nonterminal x and y from these strings are eliminated by rules r_4 and r_5, i.e.,

$$[(wa^k x, (3,19)^k)), (xb^m y, (5/19)^m] \vdash r_4$$
$$(wa^k b^m y, (3/19)^k (5/19)^m (11/19)^n).$$

Then the language generated by the probabilistic splicing system γ_2

$$L_p(\gamma_2) = \{(wa^k b^m c^n z, \tau_1^k \tau_2^k \tau_3^k) \mid k,m,n \geq 1\}$$

where $\tau_1 = 3/19$, $\tau_2 = 5/19$ and $\tau_3 = 11/19$. Further, consider the following thresholdlanguages:

$$L_p(\gamma_2 > 0) = L(\gamma_2') \in \mathbf{REG}$$

where γ_2' is the "crisp" variant of the splicing system γ_2.

$$L_p(\gamma_2, > \tau i) = \{wa^k b^m c^n z \mid 1 \leq k,m,n \leq i\} \in \mathbf{FIN}$$

where $\tau = 165/6859$, and $i \geq 1$ is a fixed positive integer.

Now, let $\Omega = \{(165/6859)^n | n \geq 1\}$, then

$$L_p(\gamma_2, \in \Omega) = \{wa^n b^n c^n z \mid n \geq 1\} \in \mathbf{CS} - \mathbf{CF}$$

and

$$L_p(\gamma_2, \notin \Omega) = \{wa^k b^m c^n z \mid k,m,n \geq 1 \wedge k \neq m, m \neq n, k \neq n\} \in \mathbf{CS} - \mathbf{CF}.$$

Two simple but interesting facts of probabilistic splicing systems state as Proposition 5.1and Proposition 5.2 below:

From Theorem 2.1, Lemma 5.1 and Example 5.1, the following two theorems are obtained:

Hence, it shows that an extension of splicing systems with probabilities increases the generative power of splicing systems with finite components, in particular cases, probabilistic splicing systems can generate non-context-free languages. Since this restricted splicing systems are unable to generate recursively enumerable languages, this area of research remains open.

6. Conclusion

In this research, some restrictions that have been imposed on splicing systems has been explored. The restrictions include weighted, groups, and probability. The definitions, theorems, and example associated with each restriction has been presented. While restricted splicing systems have been found to increase the generative power of the languages generated beyond regular languages, they still fall short of generating the highest languages, which are the recursively enumerable languages. Therefore, further research is needed to address this limitation and explore ways to increase the generative power of splicing systems towards this end.

Acknowledgements

The authors would like to thank the Ministry of Higher Education and Universiti Teknologi MARA Malaysia for the financial funding through research grant GPK-UiTM (600-RMC/GPK 5/3 (100/2020)).

REFERENCES

[1] T. Head, “Formal language theory and DNA: An analysis of the generative capacity of specific recombinant behaviors,” *Bull. Math. Biol.*, vol. 49, no. 6, pp. 737–759, 1987.

[2] S. Turaev, Y. S. Gan, M. Othman, N. H. Sarmin, and W. H. Fong, “Weighted splicing systems,” in *Communications in Computer and Information Science*, 2012, vol. 316 CCIS, pp. 416–424, 2012.

[3] N. Z. A. Hamzah, N. A. Mohd Sebry, W. H. Fong, N. H. Sarmin, and S. Turaev, “Splicing Systems over Permutation Groups of Length Two,” *Malaysian J. Fundam. Appl. Sci.*, vol. 8, no. 2, pp. 83–88, 2014.

[4] S. Turaev, M. Selvarajoo, M. H. Selamat, N. H. Sarmin, W. H. Fong, “Probabilistic splicing systems,” *Adv. Methods Comput. Collect. Intell.*, pp. 259–268, 2013.

[5] Y. Yusof, N. H. Sarmin, T. E. Goode, M. Mahmud, and W. H. Fong, “An Extension of DNA Splicing System,” *Proc. - 2011 6th Int. Conf. Bio-Inspired Comput. Theor. Appl. BIC-TA 2011*, pp. 246–248, 2011.

[6] M. Amos, G. Paun, G. Rozenberg, “Dna-based computing: a survey,” *Theor. Com- Puter Sci.*, vol. 287, no. 1, pp. 3–38, 2002.

[7] M. P. Santono, M. Selvarajoo, W. H. Fong, N. H. Sarmin, “Some Properties of Bounded-Addition Fuzzy Splicing Systems,” *Kalahari Journals*, vol. 6, no. 3, pp. 2698–2705, 2021.

[8] M. P. Santono, M. Selvarajoo, W. H. Fong, and N. H. Sarmin, “Bounded-Addition Fuzzy Simple Splicing Systems,” vol. 13, no. 2, pp. 2079–2089, 2022.

[9] H. Zhang *et al.*, “The Properties of Semi-Simple Splicing System Over Alternating Group, A3,” *iopscience.iop.org*, vol. 1770, p. 12001, 2021.

[10] M. Selvarajoo, W. H. Fong, N. H. Sarmin, and S. Turaev, “The characteristics of simple splicing languages over permutation groups,” *AIP Conf. Proc.*, vol. 2266, no. October, 2020.

[11] J. E. Hopcroft, R. Motwani, Rotwani, and J. D. Ullman, “Introduction to Automata Theory, Languages and Computability,” 2000.

[12] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*, no. January. 1997.

[13] N. H. Sarmin, Y. Yusof, and W. H. Fong, “Some characterizations in splicing systems,” *AIP Conf. Proc.*, vol. 1309, no. December, pp. 411–418, 2010.

[14] M. Selvarajoo, W. H. Fong, N. H. Sarmin, S. Turaev “Computational Power of Probabilistic Simple One-Sided Sticker Languages,” *jmcs.com.my*, vol. 2, no. 2, 2016.

[15] G. Păun, G. Rozenberg, and A. Salomaa, “DNA computing: New computing paradigms,” *Comput. Math. with Appl.*, vol. 37, no. 3, p. 134, 1998.